Toward Robust Parametric Trajectory Segmental Model for Vowel Recognition

Bing Zhao, Tanja Schultz
{bzhao, tanja}@cs.cmu.edu
Language Technologies Institute, Carnegie Mellon University

Abstract

In this paper we present a robust and discriminative segmental trajectory modeling for vowel recognition. We proposed two new approaches. One is using weighted least square estimation for the parametric trajectory parameter, which gives a much more robust performance over traditional least square estimation approach. The other is a specifically designed transformation matrix proposed to reduce the possible mismatch between the Gaussian modeling assumption and the trajectory feature's nature. Our experiments on the vowel classification using the mobile phone data of SpeechDAT(II) MDB showed significant improvement over both standard HMM and traditional segmental modeling.

1. Introduction

Vowels are generally spectrally well defined. As such they contribute significantly to our ability to recognize speech, both by human beings and speech recognizer. For vowels, the speech behavior can be considered as a point that moves in parameter space as the articulatory system changes. Standard HMM's using cepstra with their derivatives can not effectively model the trajectories especially for vowels [1][2]. In this paper we exploit robust and discriminative approaches to the parametric trajectory modeling.

Previous approaches employed linear least square estimation for the polynomial parametric trajectory [3]. This estimation could not account for the change in variance of the trajectory as a function of time and the estimation is sensitive to the large variability esp. near the phone boundaries even using multi-mixtures [4]. Instead of using least square estimation, we here proposed a new weighted least square estimation (WLS) to estimate the trajectory feature. This approach is different from the traditional approaches by giving a different weight to each frame according to its contribution to the estimation accuracy of trajectory feature. Experiments showed its robust ability in modeling the time-variation of the residual covariance and significant improvement over general linear least square estimation.

Another problem in trajectory is the high correlation within the residual error covariance, which hurts the Gaussian bayes classifier's assumption of independence. Here we proposed a transformation matrix to reduce the high correlation within the residual error covariance by transforming the features to reduce the correlation, and reduce the mismatch between the trajectory feature space and the Gaussian modeling assumptions, thus improve the performance of the segmental modeling.

It is not easy to find an answer for this transformation matrix. We present here a new discriminative approach to get this transformation matrix for segmental modeling. In this paper, we explored MCE (Minimum Classification Error [5]) training of this transformation matrix. Initializing the transformation matrix by identity matrix, we do iterative gradient search to update each element's value of the matrix. With this transformation matrix, we can tune the parametric trajectory parameter directly and can ensure the improvement of the performance. This approach avoids the difficulty of direct MCE training of the segmental model, but can still strengthen the discriminative characteristics of the feature space. Our experiments showed great improvement for trajectory modeling.

The paper continues as follows: in section 2, we will describe our weighted least square estimation for the quadratic trajectory feature extraction; in section 3, we will present our specially designed discriminative training algorithm for the transformation matrix in segmental model training; in section 4, comparisons between our approach and the traditional least square estimation, and experiments on speechDAT(II) English mobile phone database (MDB) for these approaches will be given; discussions and conclusions are given in section 5.

2. Parametric Trajectory Model

Parametric trajectory model treats each speech unit being modeled as a curve in the parametric feature space. The trajectories we are considering are vowels, and are of low degree polynomials such as quadratic polynomial followed the work in [3][4].

We model each speech segment's feature dimension as follows:

$$c(n) = \mu(n) + e(n) \text{ , for } n=1, \dots, N \quad (1)$$

Where $c(n)$ are Cepstra of a speech segment with the frame length of *N*. $\mu(n)$ is the mean feature vector representing the dynamics of features in the segment. $e(n)$ is the residual error vector, which, by assumption, has Gaussian distribution and are independent from frame to frame. The mean of the feature vector models the trajectory, which is, in our case, a quadratic function of time.

Given a speech segment of N frames, where each frame is represented with D dimensional feature vector. We model the speech segment as follows:

$$\mathrm{C} = \mathrm{ZB} + \mathrm{E} \text{ or:}$$

$$\begin{bmatrix} c_{1,1} & c_{1,2} & \dots c_{1,D} \\ c_{2,1} & c_{2,2} & \dots c_{2,D} \\ \vdots & & \\ c_{N,1} & c_{N,2} & \dots c_{N,D} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \frac{1}{N-1} & (\frac{1}{N-1})^2 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \dots \beta_{1,D} \\ \beta_{2,1} & \beta_{2,2} & \dots \beta_{2,D} \\ \beta_{3,1} & \beta_{3,2} & \dots \beta_{3,D} \end{bmatrix} + \begin{bmatrix} e_{1,1} & e_{1,2} \cdots e_{1,D} \\ e_{2,1} & e_{2,2} \cdots e_{2,D} \\ \vdots & \\ e_{N,1} & e_{N,2} \cdots e_{N,D} \end{bmatrix} \quad (2)$$

C is the feature vector matrix, which is NxD; *Z* is the NxR design Matrix that specifies the degree of the polynomials to be used, in our case *R*=3; *B* is RxD parametric trajectory matrix we are to model, and *E* is NxD residual error matrix, and assumed to be independent from frame to frame.

2.1 Estimation of the trajectory parameters

Given the notation shown above, one can now estimate the trajectory parameters under the assumption that the residual errors are independent and identically distributed [3]. By minimizing the least square objection function (3), the maximum likelihood estimation of the trajectory parameter B is as follows:

$$J_{LS}(\hat{B}) = (C - Z\hat{B})^T (C - Z\hat{B}) \quad (3)$$

$$\hat{B} = [Z^T Z]^{-1} Z^T C \quad (4)$$

$$\hat{\Sigma} = \frac{\hat{E}^T \hat{E}}{N} = \frac{(C - Z\hat{B})^T (C - Z\hat{B})}{N} \quad (5)$$

From least squares estimation, we can see that the estimation deals each training sample with equal weight. But in fact, since each observation/frame has different observation error for the trajectory, in particular of the boundaries, where large variability may cause large observation errors. It is reasonable to give more weight to the accurate observations during estimation and less weight to those samples, which has potential larger observation error.

In our approach, we employed Weighted Least Square criteria (WLS) for the estimation of the parametric trajectory. The weighted least square criteria is as follows:

$$J_{WLS}(\hat{B}) = (C - Z\hat{B})^T W (C - Z\hat{B}) \quad (6)$$

Where *W* is the weighting matrix. Now by minimizing (6), we can get the estimation for the trajectory model:

$$\hat{B}_{WLS} = [Z^T W Z]^{-1} Z^T W C \quad (7)$$

We can calculate the estimation error for the weighted linear least square estimation:

$$\tilde{B}_{WLS} = B - \hat{B}_{WLS} = -(Z^T W Z)^{-1} Z^T W E \quad (8)$$

Also assume $E[E] = 0$ and define $R = E[EE^T]$ (9), the variance of the estimation is as follows:

$$Var\tilde{B}_{WLS} = E[\tilde{B}_{WLS}\tilde{B}_{WLS}^T] = (Z^T W Z)^{-1} Z^T W E[EE^T] W Z (Z^T W Z)^{-1}$$
$$= (Z^T W Z)^{-1} Z^T W R W Z (Z^T W Z)^{-1} \quad (10)$$

Since *R* is definite positive, it can be represented as: $R = M^T M$, where *M* is a matrix, which can be inversed. Denote *A* and *B* as $A = Z^T M^{-1}$, $B = MWZ(Z^T W Z)^{-1}$, for any weighted matrix *W*, using the *Schwarz inequality* law, we get (11):

$$Var\,\tilde{B}_{WLS} = (Z^T W Z)^{-1} Z^T W R W Z (Z^T W Z)^{-1} = B^T B \geq \quad (11)$$
$$(AB)^T (AA^T)^{-1} (AB) = (AA^T)^{-1} = (Z^T R^{-1} Z)^{-1}$$

So it is now clear that when:

$$W = R^{-1} \quad (12)$$

The equation (11) is satisfied and reaches its minimal value: $Var\tilde{B}_{WLS} = (Z^T R^{-1} Z)^{-1}$. Note in the case that matrix *R* is a diagonal one, the weight is inverse proportional to the variance of the observation error, which is just what we expected: giving less weight to those samples that may have potential larger observation errors. Also the least square estimation is a special case of WLS, when W is an identity matrix.

To calculate the weighting matrix of (12), first calculate standard least square trajectory parameters estimation by (4)(5). Then with this weighting matrix *W*, we can update the estimation of both the trajectories and the residuals at all times along the trajectory.

2.2 Estimation of the Model parameters: An EM algorithm

After we have all the individual segments' trajectory parameters, the next step is to train a segmental trajectory model. Given totally *K* samples to train an *M* mixture component segmental trajectory model, the EM algorithm [4] is as follows:

Let the trajectory parameter denoted as $\{N_k, \hat{B}_k, \hat{\Sigma}_k\}$, as used in [3], given mixture component *m*, the likelihood of the segment *k* is:

$$l(\hat{B}_k, \hat{\Sigma}_k \mid B_m, \Sigma_m) = l(k \mid m) = (2\pi)^{-\frac{DN_k}{2}} |\Sigma_m|^{-\frac{N_k}{2}} \cdot \exp(-\frac{N_k}{2} tr[\Sigma_m^{-1}\hat{\Sigma}_k]) \cdot$$
$$\exp(-\frac{1}{2} tr[Z_k(\hat{B}_k - B_m)\Sigma_m^{-1}(\hat{B}_k - B_m)^t Z_k^t]) \quad (13)$$

After all $l(k \mid m)$ is calculated for totally *K* training samples, the probability of the *m* mixture given the segment *k* can be calculated:

$$p(m \mid k) = \frac{l(k \mid m) p(m)}{\sum_{i=1}^{M} l(i \mid m) p(m)} \quad (14)$$

Using ML estimates for the model parameters $p(m)$, B_m, and Σ_m:

1. Prior probability for mixture component *m*:

$$p(m) = \frac{1}{K} \sum_{k=1}^{K} p(m \mid k)$$

2. Trajectory parameter for mixture component *m*:

$$B_m = [\sum_{k=1}^{K} p(m \mid k) Z_k^T Z_k]^{-1} [\sum_{k=1}^{K} p(m \mid k) Z_k^T Z_k] \quad (15)$$

$$\Sigma_m = \frac{\sum_{k=1}^{K} p(m \mid k)(C_k - Z_k B_m)^T (C_k - Z_k B_m)}{\sum_{k=1}^{K} p(m \mid k) N_k} \quad (16)$$

The updated parameters are used to calculate $l(k \mid m)$ for next iteration of EM training.

In our approach, we used a more robust trajectory distance of the trajectory parameters to initialize the mixture components:

$$dis = tr[(C_k - Z_k B_m)^T (C_k - Z_k B_m)] \quad (17)$$

The idea of using this distance metric is based on the assumption of Gaussian modeling that the residual errors are independent and Gaussian distributed. In our experiment, we have found that using this metric is more robust and preferable to initialize each mixture component than using *l(k|m)* directly. We used *SVD* to calculate the inverse of the matrix to secure a robust EM training.

3. Estimation of Transformation Matrix

With the trajectory model trained using the WLS feature estimation and the distance of (17), we reduced the potential variations around the boundaries and gave a robust training to multi-mixture MLE model. But the fact of high correlation within the residual covariance matrix is not in accordance with the assumption of Gaussian modeling when we designed this trajectory model in section I.

In our approach, we used a transformation matrix trained under Minimum Classification Error (MCE) criteria to reduce this negative

effect. From the formula of (13), it is not easy to directly tune the parameters of B_m and Σ_m, especially for the case of full Σ_m matrix. But in fact, full matrix of Σ_m yields a much better performance than the diagonal Σ_m, and is more preferred. So we intended to design discriminative training algorithm for transformation matrix for the cases of both the diagonal and full Σ_m matrix.

Our intention is to give the parametric trajectory parameters *B* a rotation in the feature space to increase the discrimination and reduce the high within correlation in the residual covariance of the parametric feature. By this way, we can tune the likelihood of $l(k \mid m)$. Without prior knowledge about the transformation matrix, we here use discriminative criteria to get this matrix from training process. The transformation matrix is defined as follows:

$$T(\hat{B}_k^{'}) = T \cdot \hat{B}_k^{'} \tag{18}$$

T is a linear transformation. Plug in the transformation in (13), we can see the transformed likelihood is:

$$l(k \mid m) = (2\pi)^{-\frac{DN_k}{2}} \mid \Sigma_m \mid^{-\frac{N_k}{2}} \cdot \exp(-\frac{N_k}{2} tr[\Sigma_m^{-1}\hat{\Sigma}_k]) \cdot \exp(-\frac{1}{2} tr[Z_k(\hat{B}_k - B_m)T\Sigma_m^{-1}T^t(\hat{B}_k - B_m)^t Z_k^t]) \tag{19}$$

Note that if matrix T is an identity matrix, the posterior probability is the same as the original $l(k \mid m)$ in (13). The transformed likelihood of (19) showed that $l(k \mid m)$ now is not simply a function of the individual trajectory. The full covariance representing the interaction from frame to frame within the trajectory also plays an important role.

This interaction within the individual trajectory is caused by the contemporaneous correlation existing between the residuals associated with each individual trajectory respectively. And the transformation matrix T acts as a coordination function on the interaction between the individual trajectory features given the variance of the residuals associated with different features.

From the transformed likelihood of $l(k \mid m)$ of (19), we can now train the transformation matrix T via MCE. Given training sample of segment k with the reference model m_r, first calculate the best competitor m_c, we can denote the following formula for MCE training:

$$d_k = -\log[l(k \mid m_r)] + \log[l(k \mid m_c)] \tag{20}$$

Loss function is defined as:

$$L_k(d_k) = sigmoid(d_k) = \frac{1}{1 + \exp(-\gamma \cdot d_k + \theta)} \tag{21}$$

After all loss value for each training sample is calculated, we can do the MCE training for each element $\hat{T}_{uv}$ of the transformation matrix *T* as follows:

$$\hat{T}_{uv}^{(n+1)} = \hat{T}_{uv}^{(n)} - \varepsilon \frac{\partial L_k}{\partial \hat{T}_{uv}^{(n)}}$$

$$\frac{\partial L_k}{\partial \hat{T}_{uv}^{(n)}} = \frac{\partial L_k}{\partial d_k} \cdot \frac{\partial d_k}{\partial \hat{T}_{uv}^{(n)}} = \gamma \cdot L_k(d_k)(1 - L_k(d_k)) \cdot \frac{\partial d_k}{\partial \hat{T}_{uv}^{(n)}}$$
$$= \gamma \cdot L_k(d_k)(1 - L_k(d_k)) \cdot \sum_{i=1}^{M} \left[-\frac{p(m_{ri}) l(k \mid m_{ri})}{l(k \mid m_r)} \frac{\partial l(k \mid m_{ri})}{\partial \hat{T}_{uv}^{(n)}} + \frac{p(m_{ci}) l(k \mid m_{ci})}{l(k \mid m_c)} \frac{\partial l(k \mid m_{ci})}{\partial \hat{T}_{uv}^{(n)}} \right] \tag{22}$$

Plug in the transformed likelihood $l(k \mid m_{ri})$ of equation (19), it is now straightforward to calculate (22) and hence do the MCE training for each element $\hat{T}_{uv}$ of the transformation matrix. From (19) and (22), we can also see that this approach can handle with both the diagonal and the full matrix of the residual error variance.

Our MCE algorithm goes as follows:

I. Initialization: set $\hat{T}$ to be an identity matrix;
II. Scan all the training data to calculate the loss and build up the derivatives for $\hat{T}_{uv}$;
III. Using Minimum Classification Error (MCE) GPD to update each $\hat{T}_{uv}$;
IV. Using the updated transformation matrix to update the likelihood of $l(k \mid m)$ for each training sample *k*;
V. Stop when overall loss value does not change, otherwise go to step II for more iterations.

By using the transformation matrix, we update the likelihood of (19), and this discriminative approach somewhat satisfies the observation-independence of modeling assumption in section 1.

4. Experiments

All our experiments are carried out on the SpeechDAT(II) MDB. The speech data are isolated phrases such as digits, city names and application words. Using HTK3.1 [6], like [7], we built a standard monophone and a decision tree based state-tied word internal context dependent 32-mixture triphone acoustic model, which was used to do force alignment to get the vowel segments' reference. To evaluate the segmental model, we performed our experiments on a speaker independent vowel classification task. The task includes 16 vowels: /iy, ih, ey, eh, ae, aa, ah, ao, ow, uw, uh, us, er, ay, oy, aw/. We use the force alignment as the reference label and extracted the 16 vowels' training and testing tokens from the first 2 CDs. The number of training tokens extracted is 52735. The extracted testing tokens are from the test set specified by SpeechDat(II). There are totally 200 test speakers and the number of extracted test tokens is 3529. The trajectory models in this paper are context independent, full residual error covariance matrix, and quadratic polynomial trajectory model.

When doing the classification for an unknown test segment k coming from model *m* with *M* mixtures, the maximum a posterior probability rule is used as:

$$\tilde{m} = \max_m \left\{ p^{\alpha}(N_k \mid m) \sum_{i=1}^{M} L_{m_i}(\hat{B}_k, \hat{\Sigma}_k \mid \hat{B}_{m_i}, \hat{\Sigma}_{m_i}) p(m_i) \right\}$$

Where $p(N_k \mid m)$ is the duration probability that the segment *k* has frames length of N_k, and computed as a histogram during training

similar to [3], α is exponential weight experimentally set as 5.6 in experiments.

The first experiment is to compare the vowel classification between standard monophone HMMs and the parametric trajectory models using both LS and WLS estimation. The monophone HMM is trained from 5 CD's data using 13 MFCC with delta and delta delta, and the number of mixtures varies from 1 to 64. The trajectory segmental model uses 10 MFCC and 10 delta MFCC, and the number of mixtures varies from 1 to 24 mixtures. The number of the parameters referred to the number of real parameters in the model trained.

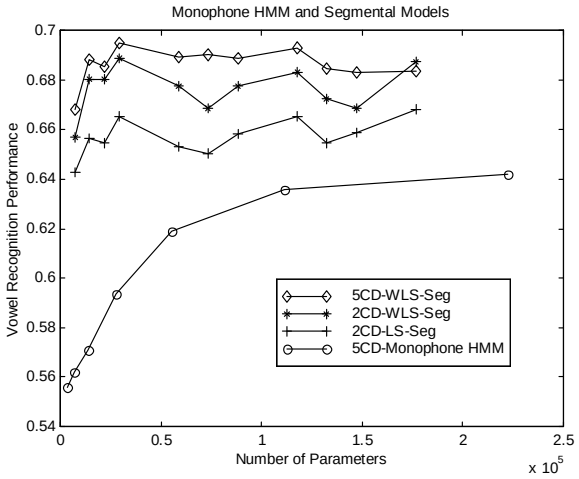

Fig 1. HMM vs WLS/LS trajectory models

From the Fig 1, we see, for this vowel classification task, WLS segmental model is better than the traditional LS segmental model, and the segmental model is more powerful than monophone HMM with the similar parameters size. Even trained with 2 CD's data the segmental model outperformed the standard monophone HMM. Second, for segmental model, when the mixture number increased over 8, under-training occurs. With this relatively small test set, the performance curve of segmental model is wavy.

The second experiment is to evaluate the effect of MCE training for the transformation matrix(s). Here we varied the number of the mixtures (Mix num) and the number of the transformations: one is an overall tied transformation matrix, and the other is 16 transformation matrixes (16-trans) for each vowel class. The result is shown in table 1:

Mix num	Baseline LS-Seg	WLS-Seg	1 Trans-WLS	16 Trans-WLS	Error reduction
1	64.30	65.68	67.16	68.09	10.62 %
2	65.63	68.04	69.37	69.28	10.22 %
3	65.43	68.04	69.71	70.78	14.99 %
4	66.51	68.89	70.98	71.52	14.03 %
8	65.29	67.72	69.17	71.61	17.70 %

Table 1. The vowel classification accuracy [%]

We here did only 5 iterations of MCE training of the transformation matrix, which cost a 900MHZ-CPU to run about 2~3 hours over the 52735 tokens. From table 1, we can see that the proposed transformation matrix to the trajectory parameters together with the weighted least square estimation is effective to improve the modeling performance by up to 17.7% error reduction over traditional least square estimation. Second, the performance drops after the number of mixtures reaches more than 8, which might be related to the relatively small training data.

The third experiment is to compare the performance of the diagonal/full error residual matrix and to see the effect of adding more training data. The result is shown in the Fig. 2:

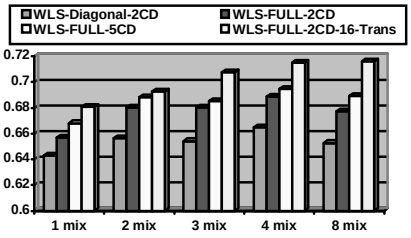

Fig 2. Performance of weighted least estimation

From Fig.2, we see that the full covariance matrix is better than the diagonal one. This is because the diagonal trajectory needs more mixtures to model feature's high variation nature. And adding more training data (5-CD, 133840 tokens) does improve the performance. And transformed WLS segmental model performed better than adding more data. We see the proposed transformation can help to satisfy the independence modeling assumption of the residual error, and improve the performance.

5. Discussion and conclusions

In this paper, we presented our approaches to improve the parametric trajectory segmental modeling power. The weighted linear estimation gives a more robust estimation of the parametric trajectory feature, and the following transformation of the feature helps reduce the mismatch between the trajectory feature's nature and the Gaussian independence assumption. Future works is to incorporate the segmental model into the framework of HMM to improve recognition performance.

6. Reference

[1]. Y. Gong, "*Stochastic Trajectory Modeling and Sentence Searching for Continuous Speech Recognition*," IEEE Trans. SAP, Vol. 5, No. 1, pp. 33-44, January 1997

[2]. W.J. Holmes and M.J. Russell. "*Linear dynamic segmental HMMs: Variability representation and training procedure.* " ICASSP, 1997.

[3]. H. Gish, Kenny Ng, "*A Segmental Speech Model with Applications to Word Spotting*", Proceedings of IEEE ICASSP, 1993.

[4]. H. Gish and K. Ng. "*Parametric trajectory models for speech recognition*", ICSLP, 1:466--469, 1996.

[5]. B.-H. Juang, W. Chou, C.-H. Lee, "*Minimum Classification Error Rate Methods for Speech Recognition*", IEEE Trans. On Speech and Audio Processing, v. 5, number 3, May 1997

[6]. HTK toolkit: http://htk.eng.cam.ac.uk/

[7]. B. Lindberg, Finn Tore Johansen, etc "*A noise robust multilingual reference recogniser based on SpeechDat(II)*" ICSLP 2000.